Methods in Molecular Biology: Insect Genomics

DOI: https://doi.org/10.1007/978-1-4939-8775-7_6

Using BUSCO to Assess Insect Genomic Resources

Robert M. Waterhouse[1,§], Mathieu Seppey[2], Felipe A. Simão[2], and Evgeny M. Zdobnov[2]

[1] Department of Ecology and Evolution, University of Lausanne, 1015 Lausanne, Switzerland.

[2] University of Geneva and Swiss Institute of Bioinformatics, 1211 Geneva, Switzerland.

[§] Correspondence should be addressed to R.M.W. robert.waterhouse@unil.ch

Summary

The increasing affordability of sequencing technologies offers many new and exciting opportunities to address a diverse array of biological questions. This is evidenced in entomological research by numerous genomics and transcriptomics studies that attempt to decipher the often complex relationships amongst different species or orders and to build 'omics' resources to drive advancement of the molecular understanding of insect biology. Being able to gauge the quality of the sequencing data is of critical importance to understanding the potential limitations on the types of questions that these data can be reliably used to address. This chapter details the use of the Benchmarking Universal Single-Copy Orthologue (BUSCO) assessment tool to estimate the completeness of transcriptomes, genome assemblies, and annotated gene sets in terms of their expected gene content.

Key Words: Genomics, Transcriptomics, Genome annotation, Completeness assessment, Single-copy orthologues

Running Title: BUSCO genomics assessments

1 Introduction

Advances in genomics technologies mean that high-throughput nucleotide sequencing has become a relatively low-cost and thus widely accessible tool with numerous applications in biological research. Nevertheless, as researchers in the field know only too well, technical issues, e.g. sample preparation, as well as biological complexities, e.g. large genome sizes, can present substantial challenges to successfully building high-quality genomics resources ***(1)***. Most leading technologies offer in-house sequencing accuracy estimates, and several

computational tools allow for detailed assessments of the performance of sequencing and assembly strategies, e.g. QUAST ***(2)*** and REAPR ***(3)*** genome assembly quality evaluators. Metrics such as contig or scaffold N50 values (half of the total assembly span is made up of contigs or scaffolds of length N50 or longer) offer a summary-statistic view of genome assembly contiguity. Scaffold counts and N50 values from a representative selection of recently published draft insect genomes show that some are currently rather fragmented and will require considerable improvement efforts to reach near-chromosomal-level status (Table 1). However, as a major goal of many genomics studies is to catalogue the complete repertoire of protein-coding genes to facilitate subsequent detailed molecular biology experiments, it is important to also assess the quality of these resources with respect to their completeness in terms of their expected gene content.

Organism	Species	Assembly size (Mbps)	Number of Scaffolds	Scaffold N50 (Kbps)	BUSCO Completeness	Publication
Fruit fly	*Drosophila serrata*	198.3	1,360	942.6	C:94.1% F:2.5%, M:1.3%	Allen *et al.* 2017 ***(4)***
Postman butterfly	*Heliconius melpomene*	275.2	795	2,103	C:81.6% F:11.1%, M:7.3%	Davey *et al.* 2016 ***(5)***
Tobacco hornworm moth	*Manduca sexta*	419.4	20,871	664	C:86.4% F:8.4%, M:5.2%	Kanost *et al.* 2016 ***(6)***
Mycalesine butterfly	*Bicyclus anynana*	475.4	10,800	638	C:98.3% F:0.9%, M:0.8%	Nowell *et al.* 2017 ***(7)***
Mediterranean fruit fly	*Ceratitis capitata*	479	1,806	4,118	C:95.6% F:3.4%, M:1.0%	Papanicolaou *et al.* 2016 ***(8)***
Bed bug	*Cimex lectularius*	650.5	1,402	7,173	C:78.6% F:14.0%, M:7.4%	Benoit *et al.* 2016 ***(9)***
Asian longhorned beetle	*Anoplophora glabripennis*	710	10,473	659	C:85.7% F:11.0%, M:3.3%	McKenna *et al.* 2016 ***(10)***
Banded demoiselle	*Calopteryx splendens*	1,630	8,896	422	C:53.5% F:31.8%, M:14.7%	Ioannidis *et al.* 2017 ***(11)***

Table 1 Assembly statistics and BUSCO assessment results from a representative selection of recently published draft insect genomes. BUSCO completeness: C, complete; F, fragmented; and M, missing. Species are ordered from the smallest to the largest assembly size and all reported values were retrieved directly from each of the publications listed.

The Benchmarking Universal Single-Copy Orthologue (BUSCO) assessment tool ***(12, 13)*** implements such quantifications of completeness for assembled genomes and transcriptomes, as well as annotated protein-coding gene sets. The assessment tool identifies matches to sets of genes that are expected to be present as single-copy orthologues in a given taxonomic group. This expectation is defined by surveying major species clades with numerous sequenced and annotated genomes to identify near-universally-present single-copy orthologues, using the ORTHODB ***(14)*** catalogue of orthologues (http://www.orthodb.org). For arthropods, BUSCO currently provides five assessment lineages: Arthropoda, Insecta, Endopterygota, Hymenoptera, and Diptera (http://busco.ezlab.org). The evolutionary filter for genes that are almost always present as single-copy orthologues across a given clade, i.e. genes evolving under 'single-copy control' ***(15, 16)***, means that they are expected to be present in any newly sequenced species from the same taxonomic group. Quantifying proportions of BUSCOS that can be reliably identified from different genomic resources therefore provides like-for-like estimates of their relative completeness that complement other quality metrics. Importantly, this means that even if a draft genome assembly is still rather fragmented, good BUSCO completeness results allow researchers to proceed with confidence knowing that they have managed to capture most of the expected protein-coding gene repertoire. The examples in Table 1 illustrate how scaffold counts or N50 values are not necessarily predictive of BUSCO completeness, highlighting the importance of such assessments to ensure transparent and intuitive genomic resource quality measures for the benefit of the entire research community.

This chapter presents step-by-step examples of using BUSCO to assess the completeness of different insect genomics resources, with sufficient detail to allow even those unfamiliar with command line computing to run their own assessments. The assessment process consists of running a computational pipeline to identify and then classify BUSCO matches from genome assemblies, annotated gene sets, or transcriptomes, using HMMER ***(17)*** hidden Markov models (HMMs). For transcriptomes the longest open reading frames are assessed, while for genome assessments, gene models are first built using *ab initio* gene prediction with AUGUSTUS ***(18)*** for the potential matches identified using TBLASTN ***(19)*** searches. Matches that meet the BUSCO HMM score cut-offs are classified as '*complete*' if their lengths fall within BUSCO profile length expectations, and if found more than once they are classified as '*duplicated*'. Those that do not meet the length requirements are considered as partial matches and are classified as '*fragmented*', and BUSCOS without matches that pass the thresholds are classified as '*missing*'. In this way, the assessments provide an intuitive quantification of the completeness of different genomics datasets in terms of expected gene content.

2 Materials

Before running BUSCO assessments, users are required to first set up the BUSCO software and its dependencies on their computer system and make sure that the data they wish to analyse adhere to the correct formats. These are outlined below, and users are encouraged to visit the website and read the user guide for further detailed information (http://busco.ezlab.org).

2.1 Software setup

1. BUSCO has been developed in Python and tested on Linux operating systems and it is therefore recommended to use a Linux machine for running BUSCO and its dependencies.
2. The software distribution is available from a public GitLab project where it can be downloaded or preferably (*see* Note 1) cloned using a git client:
   ```
   $ git clone https://gitlab.com/ezlab/busco.git
   ```
3. As well as Python, the following software packages are BUSCO dependencies and thus must also be installed on the system:
   HMMER (v3.1b2) from http://hmmer.org
   NCBI BLAST+ from https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+ (*see* Note 2)
   AUGUSTUS (v3.2.1 or above) from http://bioinf.uni-greifswald.de/augustus (*see* Note 3)
4. BUSCO v3 is installed on the system by executing the `setup.py` script (*see* Note 4):
   `$ sudo python setup.py install` (with root privileges)
   `$ python setup.py install --user` (with only user privileges)
5. BUSCO v3 setup is controlled with a user-editable configuration file. The `config.ini.default` file in the BUSCO '`config`' directory must first be copied to `config.ini` before editing. In this copied file, users must declare the paths to all dependencies (this simply tells BUSCO where they are installed on your system). Users may also use this `config.ini` file to define the input parameters for a particular analysis, but should be aware that providing input parameters through the command line will override those defined in the `config.ini` file.
6. Users without access to a Linux machine or cluster may instead use the BUSCO virtual machine (VM). The BUSCO VM was built using OSboxes (http://www.osboxes.org), it comes with the BUSCO software and its dependencies already pre-installed and can be downloaded from http://busco.ezlab.org. To run the VM, users need to first download and install a VM manager that is compatible with their system (e.g. Windows, Linux,

Macintosh, or Solaris etc.) such as VMWARE (https://www.vmware.com) or VIRTUALBOX (https://www.virtualbox.org).

7. It is highly recommended to first run a test using the sample data provided as part of the BUSCO software distribution. Execute the following commands and compare the final output '`run_TEST`' with the provided files in '`sample_data/run_SAMPLE`'.

```
$ python scripts/run_BUSCO.py --in sample_data/target.fa --out TEST
--lineage_path sample_data/example --mode genome
```

2.2 Input data

1. This chapter details the assessment of several publicly available mosquito genome assemblies and their annotated gene sets from genome sequencing projects ***(20–22)*** that can all be downloaded from VectorBase ***(23)*** (https://www.vectorbase.org/downloads).
2. The examples used for running transcriptome assessments were selected from publicly available hymenopteran datasets generated as part of large-scale insect transcriptomics studies ***(24, 25)*** that can all be downloaded from the NCBI's Transcriptome Shotgun Assembly (TSA) Sequence Database (https://www.ncbi.nlm.nih.gov/genbank/tsa).
3. Input sequence data for genome, transcriptome, or gene set assessments should be provided in standard FASTA format. Files that contain non-standard nucleotides or amino acids in the sequence lines, or non-alphanumeric or non-ASCII characters in the header lines, could cause errors and therefore these should be avoided wherever possible.
4. Pre-processing of the input data is required in order to obtain true estimates of the numbers of duplicated BUSCOs for annotated gene sets and transcriptomes, these should be pre-processed to select just one representative transcript per gene (*see* Note 5).
5. The lineage datasets used for BUSCO assessments are not provided with the software distribution. Instead, users should download the appropriate lineage dataset(s) from http://busco.ezlab.org (*see* Note 6). For example, this chapter uses genomic data from dipterans and hymenopterans, so:

```
$ wget http://busco.ezlab.org/datasets/diptera_odb9.tar.gz
$ wget http://busco.ezlab.org/datasets/hymenoptera_odb9.tar.gz
```

6. Each downloaded lineage dataset will need to be unpacked and decompressed before it can be used, for example:

```
$ tar -xf diptera_odb9.tar.gz
$ tar -xf hymenoptera_odb9.tar.gz
```

3 Methods

With the BUSCO software and its dependencies correctly set up, the relevant BUSCO lineage datasets downloaded and unpacked, and some example insect genomics datasets downloaded and pre-processed (if required), genome, gene set, and transcriptome assessments can now be performed.

3.1 Genome assessments

1. To run genome mode assessments BUSCO needs to know the location of the AUGUSTUS configuration directory so the '`config`' path must first be declared as an environment variable (*see* Note 3):

   ```
   $ export AUGUSTUS_CONFIG_PATH="/path/to/AUGUSTUS/augustus-3.2.3/config/"
   ```

2. The command to launch a genome assembly assessment is made up of four mandatory argument-value pairs that follow the python call to the `run_BUSCO.py` script:

`--in SEQUENCE_FILE`	path to your FASTA file, here your genome
`--out NAME`	a short name that identifies your analysis run
`--lineage_path LINEAGE`	path to the BUSCO lineage dataset directory
`--mode MODE`	specify which analysis mode to run, here 'genome'

   So to launch an assessment of the genome assembly of the *Anopheles arabiensis* mosquito using the dipteran lineage dataset the command would be (*see* Note 7):

   ```
   $ python /path/to/busco/scripts/run_BUSCO.py
   --in /path/to/mosquito/genomes/Anopheles-arabiensis-D1-genome.fs
   --out AARAD1
   --lineage /path/to/lineage/dataset/diptera_odb9
   --mode genome
   ```

3. There are several additional argument-value pairs that are optional and allow users to change the default values of various settings, e.g. if the user's system has access to multiple computing cores then they can take advantage of this using the `--cpu` argument (CPU, central processing unit), or the e-value cut-off for TBLASTN searches can be changed with the `--evalue` argument (*see* Note 8).
4. One of the most important optional arguments to consider for genome assembly assessments is the choice of AUGUSTUS pre-trained species-specific gene prediction parameters. Each BUSCO lineage dataset has a predefined default selection, e.g. for

the Diptera lineage the default is '`fly`', which are AUGUSTUS gene prediction parameters pre-trained on the fruit fly, *Drosophila melanogaster* (*see* Note 9).

5. Running the above assessment of the 246.6 megabasepair (Mbp) *Anopheles arabiensis* genome assembly using the dipteran lineage dataset on 12 CPUs with otherwise default options should take approximately four hours (*see* Note 10).

3.2 Gene set assessments

1. Launching an assessment of an annotated gene set follows the same basic rules as for genome assemblies, with the same four mandatory argument-value pairs that follow the python call to the `run_BUSCO.py` script:

`--in SEQUENCE_FILE`	path to your FASTA file, here your proteins
`--out NAME`	a short name that identifies your analysis run
`--lineage_path LINEAGE`	path to the BUSCO lineage dataset directory
`--mode MODE`	specify which analysis mode to run, here 'proteins'

So to launch an assessment of the *Anopheles arabiensis* annotated gene set (version AaraD1.6) using the dipteran lineage dataset the command would be:

```
$ python /path/to/busco/scripts/run_BUSCO.py
--in /path/to/mosquito/genesets/Anopheles-arabiensis-D1-proteins-1.6.fs
--out AARAD16
--lineage /path/to/lineage/dataset/diptera_odb9
--mode proteins
```

2. Running this assessment of the 13,452 *Anopheles arabiensis* (AaraD1.6) protein-coding genes using the dipteran lineage dataset on 4 CPUs with otherwise default options should take approximately forty minutes (*see* Notes 10,11).

3.3 Transcriptome assessments

1. Transcriptome assessments are launched with the same four mandatory argument-value pairs that follow the python call to the `run_BUSCO.py` script:

`--in SEQUENCE_FILE`	path to your FASTA file, here your transcripts
`--out NAME`	a short name that identifies your analysis run
`--lineage_path LINEAGE`	path to the BUSCO lineage dataset directory
`--mode MODE`	specify which analysis mode to run, here 'transcriptome'

So to launch an assessment of the transcriptome from an adult *Pelecinus polyturator* parasitoid wasp (NCBI BioProject: PRJNA252202) using the hymenopteran lineage dataset the command would be:

```
$ python /path/to/busco/scripts/run_BUSCO.py
--in /path/to/wasp/transcriptomes/Pelecinus_polyturator.fs
--out PPOLY
--lineage /path/to/lineage/dataset/hymenoptera_odb9
--mode transcriptome
```

2. Running this assessment of the 35,969 *Pelecinus polyturator* transcripts using the hymenopteran lineage dataset on 4 CPUs with otherwise default options should take approximately two hours and forty minutes (*see* Note 10). For this analysis the transcriptome was not pre-processed to remove highly-similar transcripts (*see* Note 5).

3.4 Understanding the results

1. Successful assessments will each produce a simple summary results file that reports the full command used to launch the assessment (this is useful in order to be able to re-run the same analysis), as well as the percentages and counts of '*complete*' (single-copy and duplicated), '*fragmented*', and '*missing*' BUSCOs. For example, the results of the *Anopheles arabiensis* genome assembly assessment:

```
C:98.2%[S:98.1%,D:0.1%],F:0.7%,M:1.1%,n:2799
2750   Complete BUSCOs (C)
2746   Complete and single-copy BUSCOs (S)
4      Complete and duplicated BUSCOs (D)
19     Fragmented BUSCOs (F)
30     Missing BUSCOs (M)
2799   Total BUSCO groups searched
```

2. All three assessment modes will also produce a '`full_table`' file with classification results for each BUSCO, a '`missing_busco_list`' file with the missing BUSCOs, and a '`hmmer_output`' directory with the full results of the HMM searches. In addition, genome and transcriptome mode assessments will report the results of TBLASTN searches, and genome mode results include the details for all the AUGUSTUS gene predictions as well as AUGUSTUS training parameters and the nucleotide and protein sequences of the identified complete single-copy BUSCOs (*see* Note 12).
3. The BUSCO plotting tool enables users to visualise their results as a simple bar chart, allowing for clear comparisons of different datasets (*see* Note 13). To generate a chart, users must first copy the short summary results files from each assessment that they want

to visualise into a single directory. The `generate_plot.py` script can then be launched pointing to this directory to automatically produce the chart. For example, the commands below will plot the results from the *Anopheles arabiensis* genome (AARAD1) and gene set (AARAD16) assessments, producing the image file `busco_figure.png` in the same directory where the summary results were copied:

```
$ mkdir arabiensis_results
$ cp run_AARAD1/short_summary_AARAD1.txt arabiensis_results/.
$ cp run_AARAD16/short_summary_AARAD16.txt arabiensis_results/.
$ python /path/to/busco/scripts/generate_plot.py -wd arabiensis_results/
```

4. Repeating the steps outlined above to assess a total of 15 publicly available mosquito genome assemblies and their annotated gene sets and then plotting the results enables the like-for-like comparison of these genomic resources, where all but five datasets are more than 95% '*complete*' (Figure 1). Furthermore, these mosquito genomics resources all show very low levels of duplications, indicating that the assemblies are likely mostly free of haplotype regions (*see* Note 14). In addition, the genome assembly results generally mirror those of the gene sets, with the assemblies usually performing slightly better apart from a few cases where the assembly appears substantially better than the gene set (*see* Note 15). These estimates of '*complete*', '*fragmented*', and '*missing*' BUSCOs (*see* Note 16) provide intuitive metrics with which to gauge the relative quality of these genomic resources in terms of their expected gene content.
5. Repeating the steps outlined above for the assessments of many more publicly available hymenopteran transcriptomes and plotting the resulting completeness scores against the numbers of transcripts demonstrates their highly variable completeness (Figure 2). Transcriptomes may well be expected to show rather variable completeness scores as the total repertoire of RNAs that are sequenced and assembled will often reflect the type of biological sample, e.g. a pooled sample from multiple tissues and life-stages will probably capture more than a sample from a specialised tissue (*see* Note 17).

4 Notes

1. Users are encouraged to use the git client option to retrieve the BUSCO software as this will make installation of future updates much simpler and easy to manage. Additionally,

the GitLab project '*issues*' page is worth consulting as it is a good source of tips and discussions from BUSCO users.

2. It has been reported that when running BUSCO using multiple cores, the TBLASTN step from BLAST+ versions 2.4, 2.5, and 2.6 may occasionally fail to complete and thus the BUSCO assessment will fail with an error message to this effect. To avoid this problem, use an earlier BLAST+ version or run using only a single core.
3. Users only need to install AUGUSTUS if they plan to assess genome assemblies. As AUGUSTUS has dependencies of its own, e.g. Perl, users should consult the AUGUSTUS documentation for the correct installation procedures. If working on a system where AUGUSTUS has already been installed by an administrator and the user does not have 'write permission' to the AUGUSTUS '`config`' directory, users can simply recursively copy the entire '`config`' directory to a location where they do have 'write permission' and then re-set the '`config`' path variable to this location:

```
$ cp -r /path/to/AUGUSTUS/augustus-3.2.3/config /my/home/augustus/config
$ export AUGUSTUS_CONFIG_PATH="/my/home/augustus/config/"
```

4. This was not a requirement for BUSCO v1 or v2. The v3 update refactored the underlying analysis code to make it more modular and extendable and thus it must be installed using the `setup.py` script.
5. For annotated gene sets the transcript-to-gene relationships are defined in the annotation files, e.g. General Feature Format (GFF) files, so the longest protein-coding transcript can be selected for each gene with multiple annotated transcripts. For *de novo* transcriptomes, i.e. those without a reference genome, transcript-to-gene relationships are not defined so users have two options: (i) run the assessments without pre-processing and acknowledge the fact that estimates of duplicated BUSCOs are likely to be inflated by the presence of multiple transcripts from the same gene, or (ii) pre-process the transcriptome with a sequence identity (or similarity) and length filter to select just one representative from sets of highly similar transcripts, e.g. using CD-HIT ***(26)***.
6. There are currently 16 bacterial lineage datasets and 28 eukaryotic lineage datasets. Users would normally select the most specific lineage available, i.e. the most recent ancestor of the species whose data is to be assessed. For example, for assessing ant data one would select the '`hymenoptera`' lineage rather than the '`arthropoda`' lineage. However, if there are a large number of species/strains/versions etc. to be assessed then to minimise runtime (at the expense of resolution) one might select a less specific (more ancestral) lineage dataset with fewer BUSCOs, at least for the initial rounds of assessments.
7. BUSCO outputs the running log details to the default standard output (user's terminal), in order to instead send these details to a file users can simply end the launch command

with a redirect command, and as BUSCO assessments can take some time it is useful to run them in the background, i.e. end launch command with: `>& my_log_file.txt &`

8. The optional arguments for launching a BUSCO assessment give the user flexibility over many aspects, some specific to running in genome mode and others applicable in any mode, all of which are described in full in the user guide. Some useful options to consider employing include (i) `--force`, this will force the results to overwrite results from an analysis run with the same name (ii) `--tarzip`, this will package and compress the results from steps that can produce many output files; (iii) `--augustus_options`, this allows users to pass AUGUSTUS-specific parameters for gene prediction, e.g. to use alternative codon translation tables.
9. AUGUSTUS comes with pre-trained gene prediction parameters for many species (see AUGUSTUS documentation for up-to-date details), so if parameter sets are available for the species to be assessed then they should be selected e.g. for the Florida carpenter ant, the parameter set to use can be specified by adding the argument '`--species camponotus_floridanus`' to the launch command. For many other species, pre-trained gene prediction parameters are not yet available so users should select the closest species for which such parameters are available, or run the assessment with the pre-selected default parameters. For the sake of reproducibility it is important to specify which one was selected when reporting BUSCO results.
10. Assessment runtimes will vary according to the exact system setup. Assessments of genome assemblies require the initial steps of first identifying genomic regions that potentially harbour BUSCO matches and then predicting gene models in these regions. These are computationally intensive tasks and therefore genome assembly assessments will take substantially longer than transcriptome or gene set assessments. Note also that the searches and gene predictions are performed in two rounds: (i) searches with consensus sequences built from BUSCO HMMs followed by gene predictions using the selected AUGUSTUS pre-trained parameter set; (ii) then for BUSCOs that were classified as '*fragmented*' or '*missing*' after the first round, searches with variant consensus sequences followed by gene predictions using parameters trained on '*complete*' BUSCOs identified in round one. Thus if the first round identifies a high proportion of '*complete*' BUSCOs then the second round will be relatively quick, but if there are many '*fragmented*' or '*missing*' BUSCOs after the first round then the second round will take considerably longer.
11. The *Anopheles arabiensis* AaraD1.6 annotation contains 13,452 protein-coding genes with 13,640 transcripts so the protein FASTA file downloaded from VectorBase was first pre-processed to select the longest protein per gene. Performing this pre-processing step on annotated gene sets is not obligatory, but it ensures that BUSCO estimates of the

numbers of duplicated genes will be true assessments that are not inflated by alternative transcripts that would be reported as gene duplicates.

12. During genome assessments the second round of gene predictions uses parameter sets built from the '*complete*' BUSCOs identified in the first round. These AUGUSTUS retraining parameters are saved in the '`augustus_output`' results directory. They are ideal for use during whole genome annotation procedures that employ AUGUSTUS, especially when parameter sets for the species to be annotated or those of a close relative are not already available. In addition, the GenBank or GFF formatted '*complete*' BUSCO annotations provided in the results directory can be used to train other gene predictors, e.g. SNAP ***(27)***.
13. The BUSCO plotting tool uses R (https://www.r-project.org) and the GGPLOT2 library (http://ggplot2.org), so these must be installed and accessible on the system in order to produce the image. Alternatively, adding the optional argument `--no_r` to the command will simply produce the R script required to build the image and users can then run this R script on any system where R and GGPLOT2 are installed. This also gives the user the opportunity to edit the R script to tailor the resulting image, e.g. changing the default fonts, labels, or colours etc.
14. If high levels of complete duplicates are reported for a genome assembly then this could suggest that the assembly procedure has failed to correctly collapse haplotype regions, resulting in numerous pairs of highly-similar duplicate gene copies. This would warrant further investigations to determine if this is indeed the case and if alternative assembly strategies need to be employed or if such regions can be removed or collapsed. However, knowledge of the biology of the sample itself can also offer explanations: e.g. assessing the *Aedes albopictus* C6/36 cell line genome assembly and annotated gene set suggested that most BUSCOs were duplicated, but cytogenetic studies have shown that this cell line does have aberrant karyotypes, which could explain the numerous duplicates ***(28)***.
15. Differences in the results from assessing a genome assembly versus its annotated gene set may be due to several factors. In both cases BUSCO attempts to classify the matches to a set of protein-coding gene annotations: for genomes these annotations are built by BUSCO using AUGUSTUS gene predictions with BUSCO HMMs, whereas for gene sets they have usually been built by genome annotation pipelines (e.g. MAKER ***(29)***) that incorporate evidence from several gene predictors and different sources of gene model support. Thus when results from assembly assessments appear to be better than for their gene sets it suggests that the targeted approach taken by BUSCO has produced better gene models than a more generalist annotation pipeline (at least for the subset of genes that make up the BUSCO lineage dataset). Conversely, if a gene set appears more complete than its genome this suggests that the multiple sources of evidence used by the

annotation pipeline have resulted in generally better annotations than the single-predictor approach taken by BUSCO.

16. When interpreting BUSCO results, users should be aware that the classification procedure (described in the introduction) results in the labels '*complete*', '*fragmented*', and '*missing*', which are by necessity simplifications that reflect the most likely scenario. For example, the label '*missing*' is applied to BUSCOs with no matches (probably truly absent from the dataset), but also to matches that do not meet the HMM score cut-offs. These below-cut-off matches could mean that these BUSCOs are in fact partially present in the dataset but there is simply not enough matching sequence be confident of the partial match and classify them as '*fragmented*'.
17. BUSCO assessments are usually performed to demonstrate the good completeness levels of the genomic resources generated and analysed in a particular study. However, if the aim of a transcriptomics experiment is to sample a specific tissue or life-stage where the repertoire of transcripts is expected to be highly specialised, then low completeness scores would in fact offer support that such targeted sampling was successful.

Acknowledgement

R.M.W. was supported by Swiss National Science Foundation award PP00P3_170664.

References

1. S. Richards and S.C. Murali (2015) Best practices in insect genome sequencing: What works and what doesn't, Current Opinion in Insect Science. 7, 1–7.
2. A. Gurevich, V. Saveliev, N. Vyahhi, et al. (2013) QUAST: quality assessment tool for genome assemblies, Bioinformatics. 29, 1072–1075.
3. M. Hunt, T. Kikuchi, M. Sanders, et al. (2013) REAPR: a universal tool for genome assembly evaluation, Genome Biol. 14, R47.
4. S.L. Allen, E.K. Delaney, A. Kopp, et al. (2017) Single-Molecule Sequencing of the Drosophila serrata Genome, G3: Genes, Genomes, Genetics. 7, 781–788.
5. J.W. Davey, M. Chouteau, S.L. Barker, et al. (2016) Major Improvements to the Heliconius melpomene Genome Assembly Used to Confirm 10 Chromosome Fusion Events in 6 Million Years of Butterfly Evolution., G3 (Bethesda, Md.). 6, 695–708.
6. M.R. Kanost, E.L. Arrese, X. Cao, et al. (2016) Multifaceted biological insights from a draft genome sequence of the tobacco hornworm moth, Manduca sexta, Insect

Biochemistry and Molecular Biology. 76, 118–147.
7. R.W. Nowell, B. Elsworth, V. Oostra, et al. (2017) A high-coverage draft genome of the mycalesine butterfly Bicyclus anynana, GigaScience. 6, 1–7.
8. A. Papanicolaou, M.F. Schetelig, P. Arensburger, et al. (2016) The whole genome sequence of the Mediterranean fruit fly, Ceratitis capitata (Wiedemann), reveals insights into the biology and adaptive evolution of a highly invasive pest species, Genome Biology. 17, 192.
9. J.B. Benoit, Z.N. Adelman, K. Reinhardt, et al. (2016) Unique features of a global human ectoparasite identified through sequencing of the bed bug genome, Nature Communications. 7, 10165.
10. D.D. McKenna, E.D. Scully, Y. Pauchet, et al. (2016) Genome of the Asian longhorned beetle (Anoplophora glabripennis), a globally significant invasive species, reveals key functional and evolutionary innovations at the beetle–plant interface, Genome Biology. 17, 227.
11. P. Ioannidis, F.A. Simao, R.M. Waterhouse, et al. (2017) Genomic features of the damselfly Calopteryx splendens representing a sister clade to most insect orders, Genome Biology and Evolution. 9, 415–430.
12. F.A. Simão, R.M. Waterhouse, P. Ioannidis, et al. (2015) BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs, Bioinformatics. 31, 3210–3212.
13. R.M. Waterhouse, M. Seppey, F.A. Simão, et al. (2017) BUSCO applications from quality assessments to gene prediction and phylogenomics, Molecular Biology and Evolution.
14. E.M. Zdobnov, F. Tegenfeldt, D. Kuznetsov, et al. (2017) OrthoDB v9.1: Cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs, Nucleic Acids Research. 45, D744–D749.
15. R.M. Waterhouse, E.M. Zdobnov, and E. V. Kriventseva (2011) Correlating traits of gene retention, sequence divergence, duplicability and essentiality in vertebrates, arthropods, and fungi, Genome Biology and Evolution. 3, 75–86.
16. R.M. Waterhouse (2015) A maturing understanding of the composition of the insect gene repertoire, Current Opinion in Insect Science. 7, 15–23.
17. S.R. Eddy (2011) Accelerated Profile HMM Searches, PLoS Comput Biol. 7, e1002195.
18. O. Keller, M. Kollmar, M. Stanke, et al. (2011) A novel hybrid gene prediction method employing protein multiple sequence alignments, Bioinformatics. 27, 757–763.
19. C. Camacho, G. Coulouris, V. Avagyan, et al. (2009) BLAST+: architecture and applications, BMC Bioinformatics. 10, 421.
20. R.A. Holt, G.M. Subramanian, A. Halpern, et al. (2002) The genome sequence of the

malaria mosquito Anopheles gambiae., Science. 298, 129–49.
21. X. Jiang, A. Peery, A.B. Hall, et al. (2014) Genome analysis of a major urban malaria vector mosquito, Anopheles stephensi., Genome biology. 15, 459.
22. D.E. Neafsey, R.M. Waterhouse, M.R. Abai, et al. (2015) Highly evolvable malaria vectors: The genomes of 16 Anopheles mosquitoes, Science. 347, 1258522–1258522.
23. G.I. Giraldo-Calderón, S.J. Emrich, R.M. MacCallum, et al. (2015) VectorBase: an updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases, Nucleic Acids Res. 43, D707-13.
24. R.S. Peters, L. Krogmann, C. Mayer, et al. (2017) Evolutionary History of the Hymenoptera., Current biology : CB. 27, 1013–1018.
25. M. Petersen, K. Meusemann, A. Donath, et al. (2017) Orthograph: a versatile tool for mapping coding nucleotide sequences to clusters of orthologous genes., BMC bioinformatics. 18, 111.
26. W. Li and A. Godzik (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences., Bioinformatics. 22, 1658–1659.
27. I. Korf (2004) Gene finding in novel genomes, BMC Bioinformatics. 5, 59.
28. R.M. Waterhouse, X. Chen, M. Bonizzoni, et al. (2017) The third International Workshop on Aedes albopictus: building scientific alliances in the fight against the globally invasive Asian tiger mosquito, Pathogens and Global Health. 111, 161–165.
29. M.S. Campbell, C. Holt, B. Moore, et al. (2014) Genome Annotation and Curation Using MAKER and MAKER-P., Current protocols in bioinformatics. 48, 4.11.1-39.

Figures

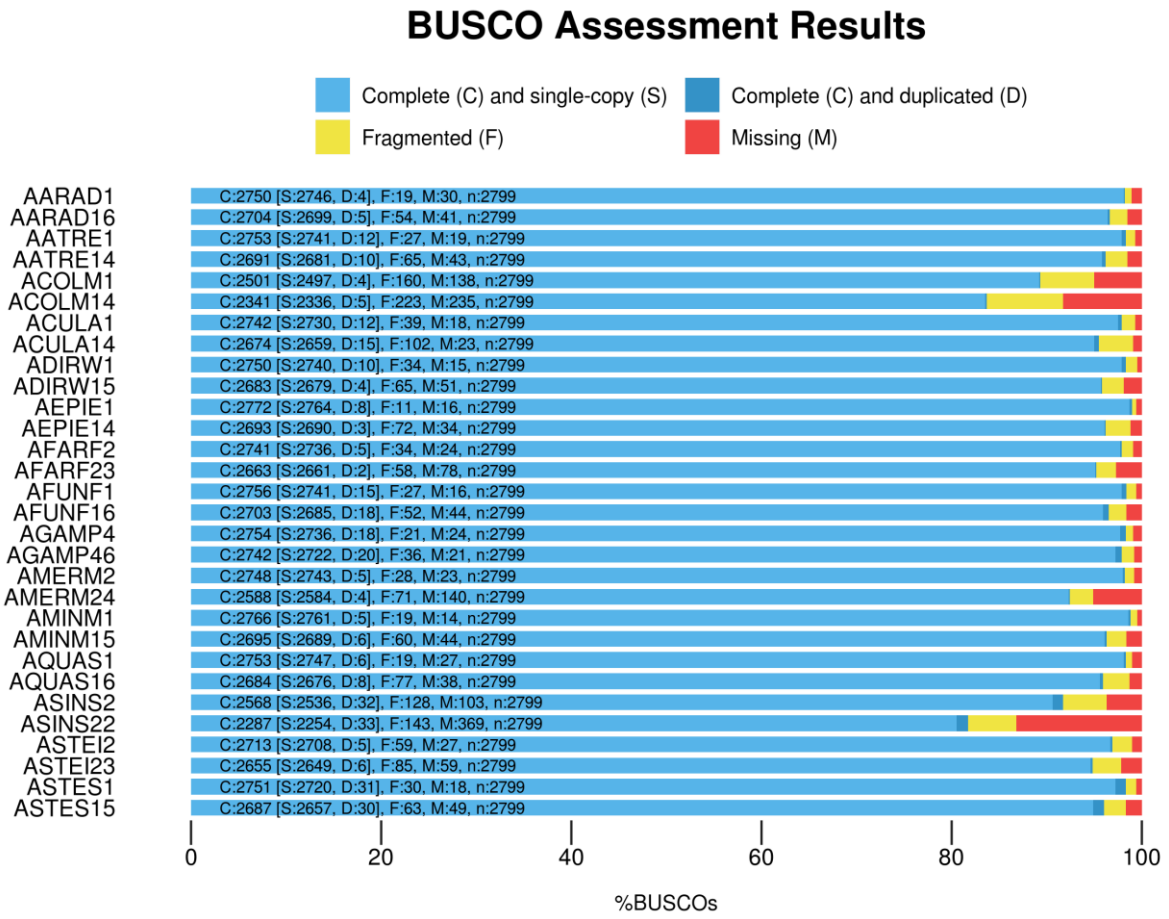

Fig. 1 BUSCO results from the assessments of 15 *Anopheles* mosquito genomes and their annotated gene sets. The chart was produced using the BUSCO plotting tool and demonstrates the intuitive visualisation of side-by-side genome and gene set results from multiple species. Gene set results (e.g. AARAD16) are plotted directly beneath genome assembly results (e.g. AARAD1) for each species with numbers indicating dataset versions: *An. arabiensis* (AARAD), *An. atroparvus* (AATRE), *An. coluzzii* (ACOLM), *An. culicifacies* (ACULA), *An. dirus* (ADIRW), *An. epiroticus* (AEPIE), *An. farauti* (AFARF), *An. funestus* (AFUNF), *An. gambiae* (AGAMP), *An. merus* (AMERM), *An. minimus* (AMINM), *An. quadriannulatus* (AQUAS), *An. sinensis* (ASINS), *An. stephensi* Indian (ASTEI), *An. stephensi* SDA-500 (ASTES).

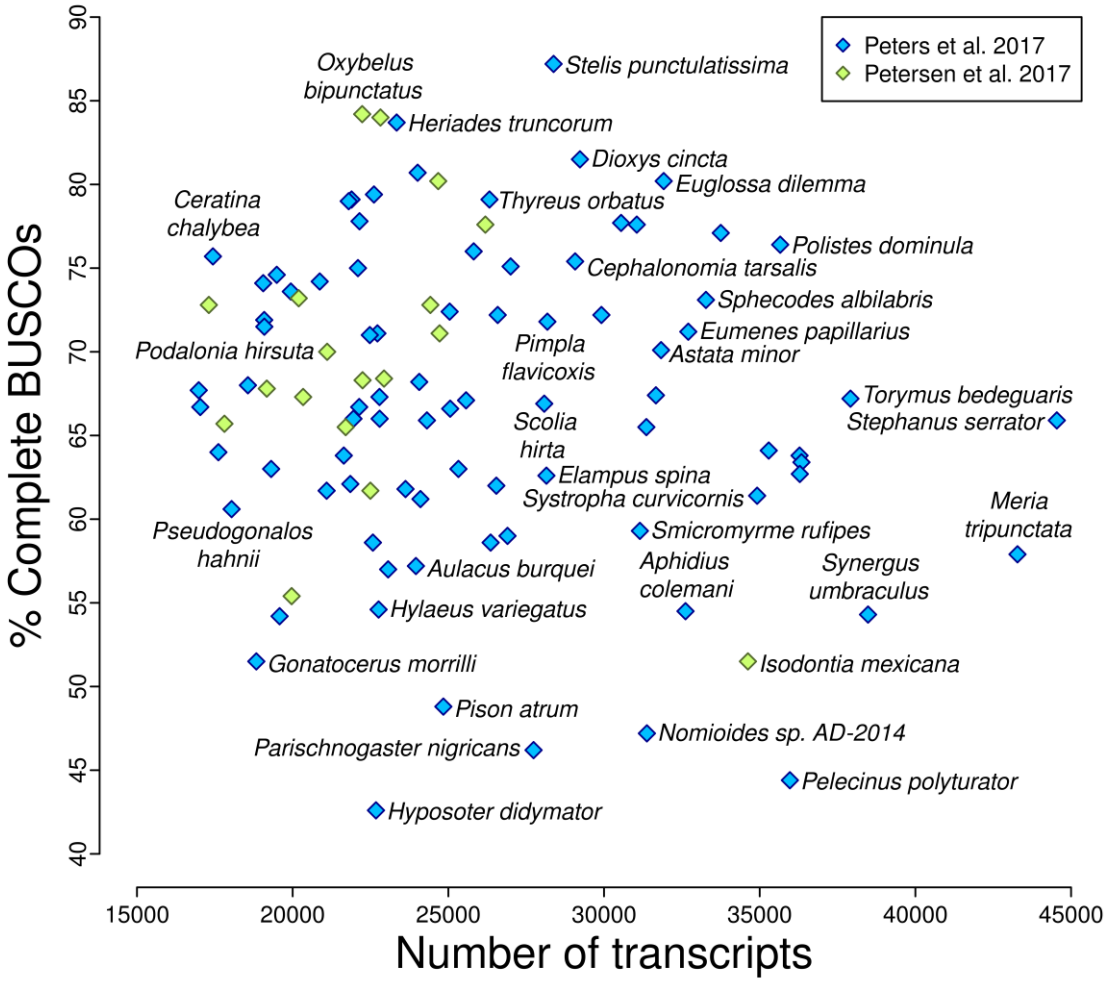

Fig. 2 BUSCO completeness results from the assessments of 103 hymenopteran transcriptomes from Peters *et al*. 2017 ***(24)*** and Petersen *et al*. 2017 ***(25)*** compared to the number of transcripts in each transcriptome. Transcriptomes with many transcripts are not necessarily the most complete, and those with fewer transcripts can still score relatively well in terms of completeness. Several example species are labelled either directly to the left or right of the data point or centred directly above or below it.